INTEGRATING FORMAL METHODS INTO SOFTWARE DEPENDABILITY ANALYSIS

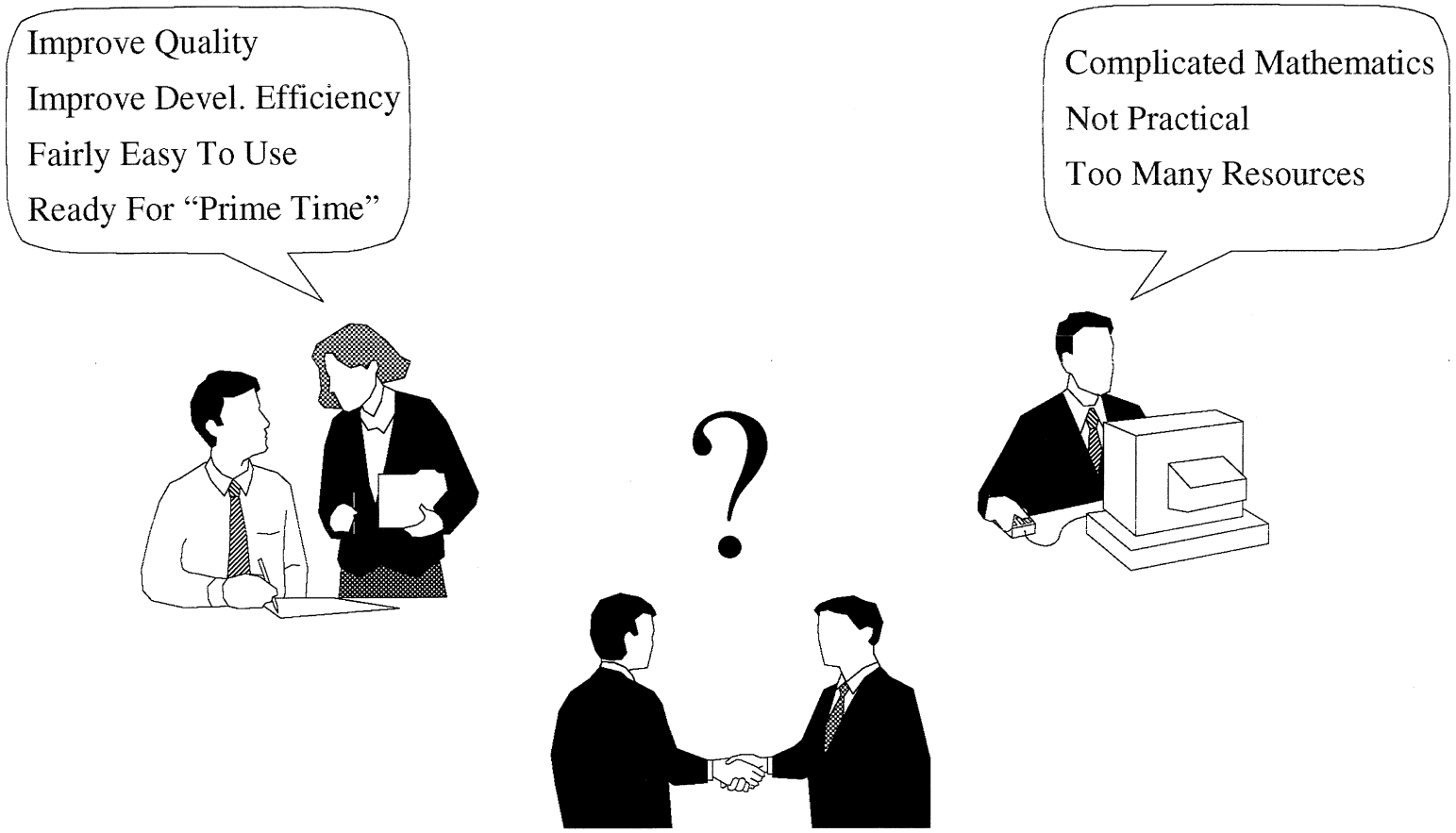
John C. Knight and Luís G. Nakano

Department of Computer Science
University of Virginia
Charlottesville
Virginia, USA

UVA

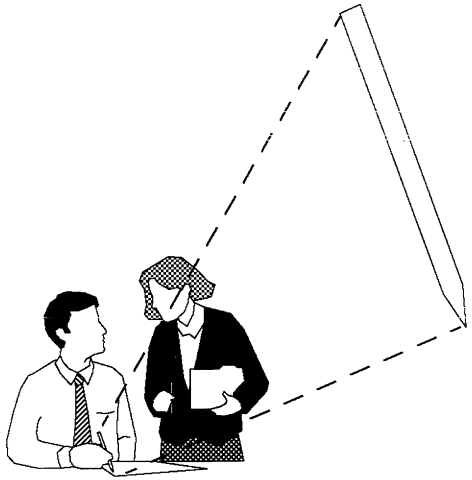
Department of Computer Science

THE USE OF FORMAL METHODS



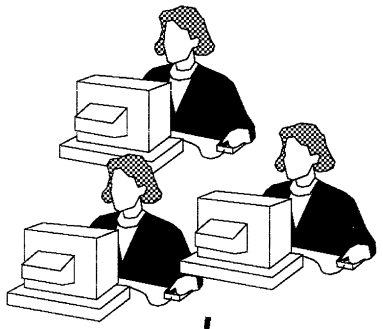
UVA
Department of Computer Science

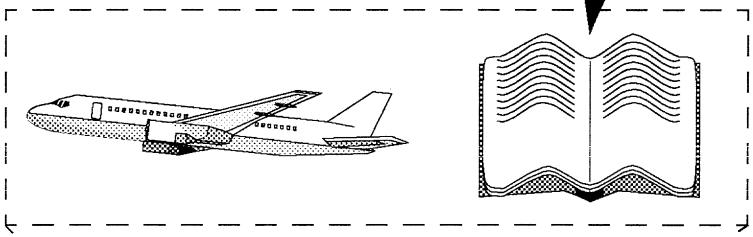
HYPOTHESES

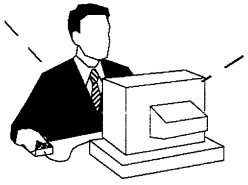


Formal
Techniques

Software
Development
Process







System
Dependability
Assessment

UVA
Department of Computer Science

QUESTION

- There Are Many Excellent Formal Techniques, Including:
  - Formal Specification
  - Specification Analysis
  - Refinement/Reification
  - Correctness Proofs
  - Property Proofs
- But, Comprehensive Formal Analysis Of Large Systems Is Impractical, So:

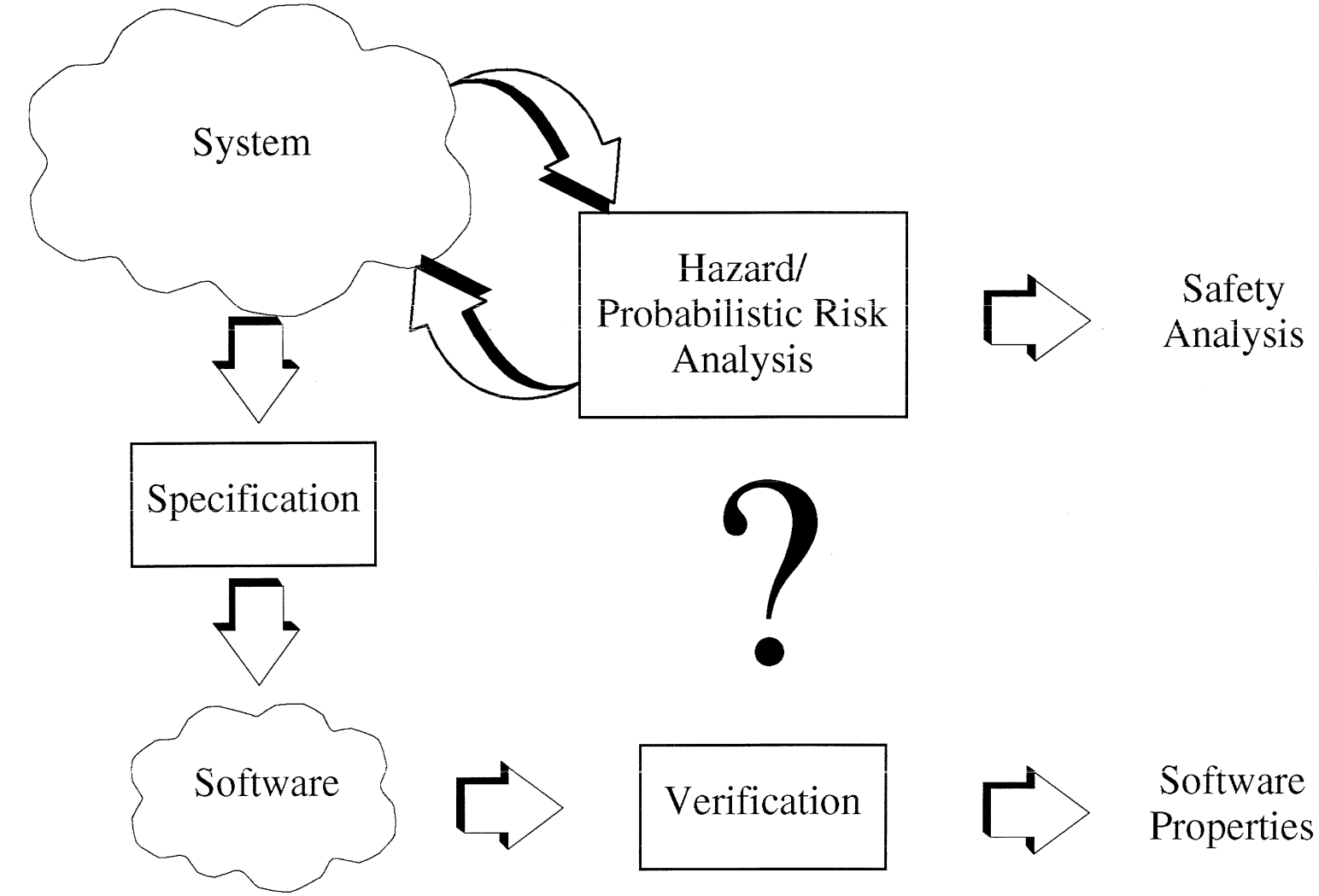
To which parts of a system should formal techniques be applied?

- Need A Means Of Determining Where They Can Be Applied Most Effectively

APPLICATION AND EVALUATION

- Formal Specification Has Been Used Extensively, For Example:
  - Statecharts — By Airbus, Guidant, Boeing
  - Z — By IBM, Praxis
  - SCR/Core — By NRL, Lockheed
  - PVS — By JPL, Rockwell Collins
- Various Evaluations Performed, For Example:
  - Craigen, Gerhart, Ralston (NIST)
  - Ardis et al. (Lucent)
- Previous Work Did Not Address Breadth Of Use:
  - Evaluation Criteria Tended To Be Technical, Performance Oriented

WHY BE FORMAL IN SW DEVELOPMENT?

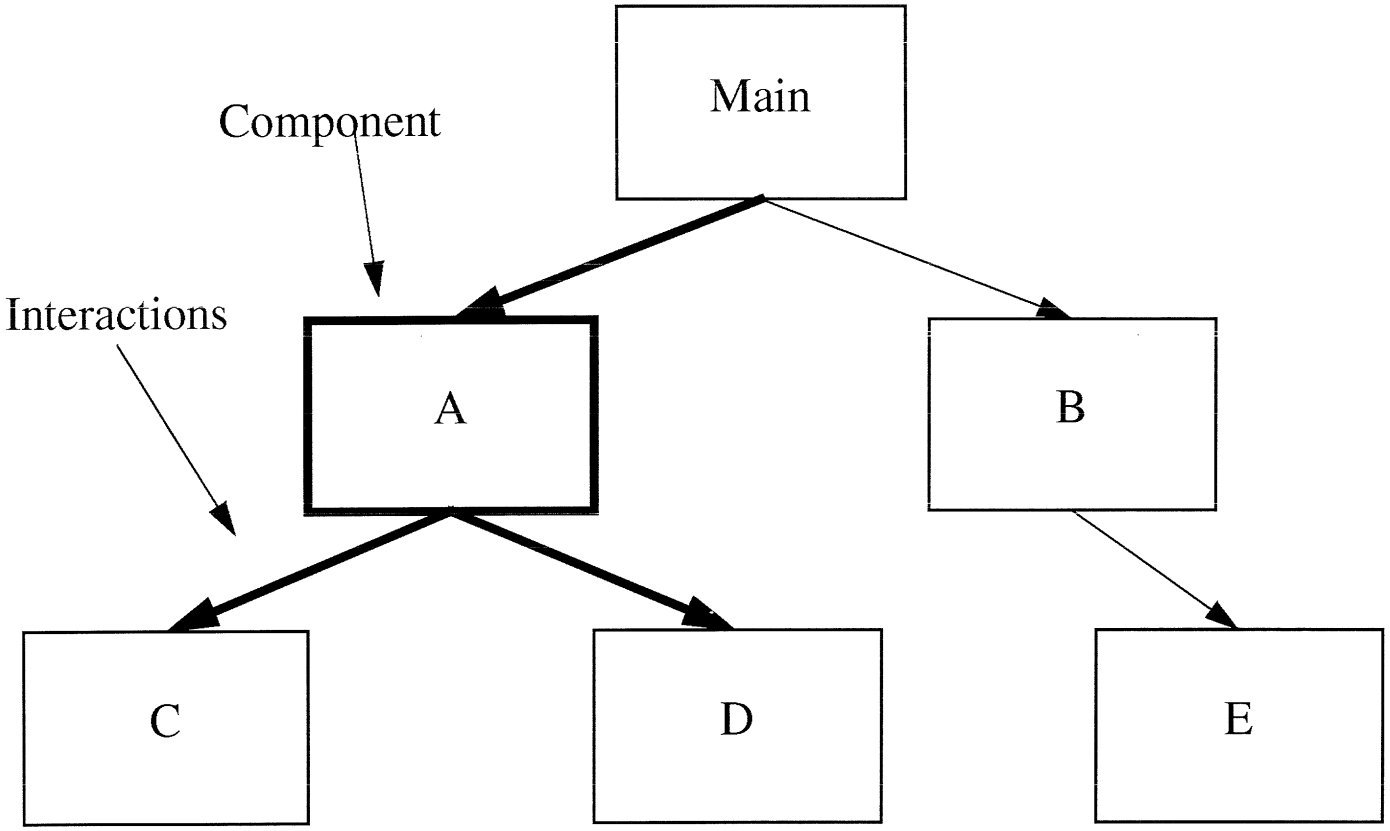


UVA
Department of Computer Science

SW IN SYSTEM DEPENDABILITY ASSESSMENT

- Software Provides Lots Of Functions:
  - Are They All Critical?
  - Do They All Fail The Same Way?
- Typical Practices—Assume Only One Software Failure Event, And:
  - Try To Measure Probability Of Failure By Life Testing
  - Or Set Probability Of Failure To One
  - Or Maybe Zero
  - Or Maybe Model Using A "Reasonable" Distribution
- Software System Life Testing:
  - Its Generally Infeasible (Butler And Finelli)
  - Its Worse Than That (Ammann, Brilliant, And Knight)

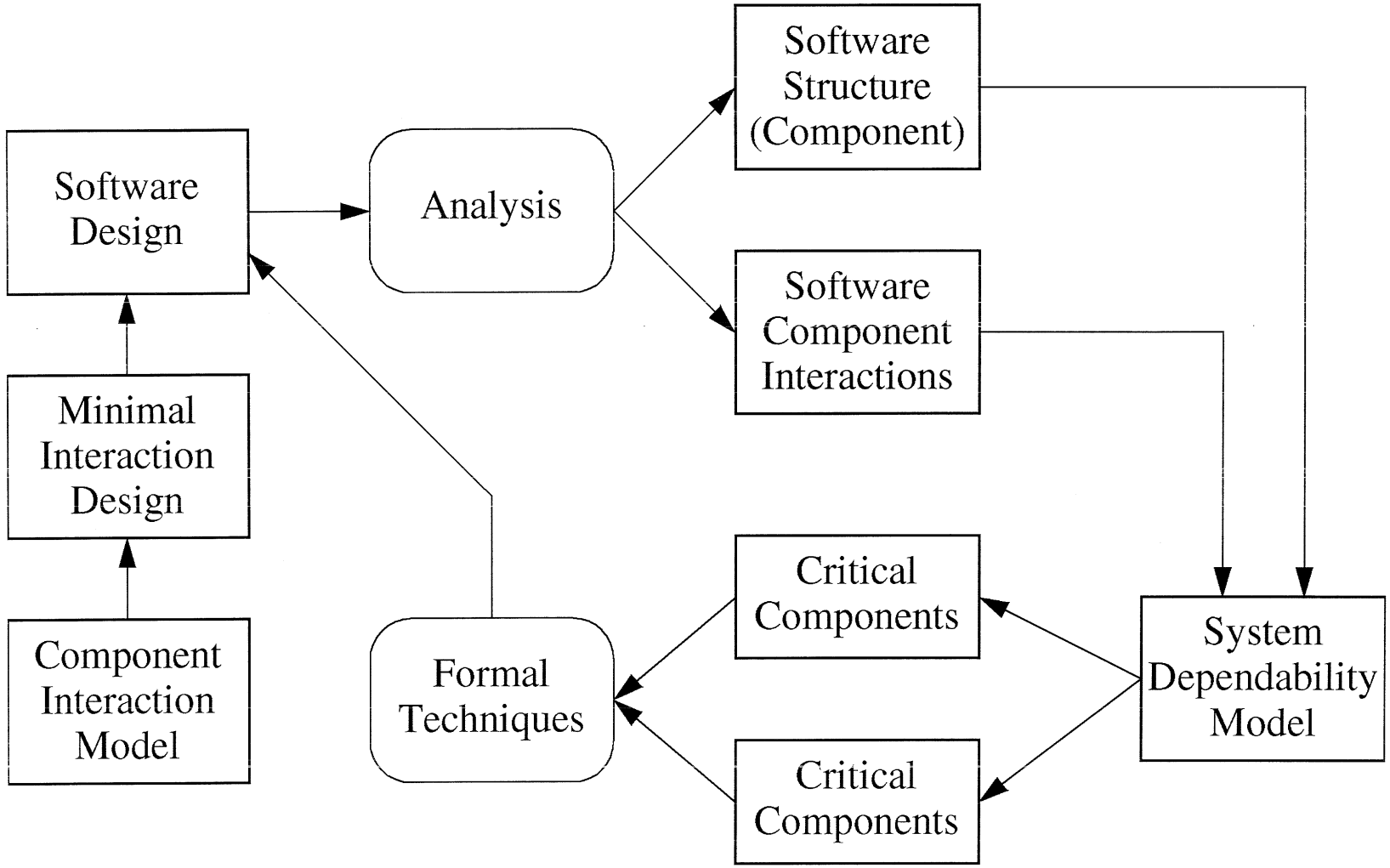
SOFTWARE STRUCTURE



- Model Software Using Its Component Structure
- Analogy With Hardware Analysis

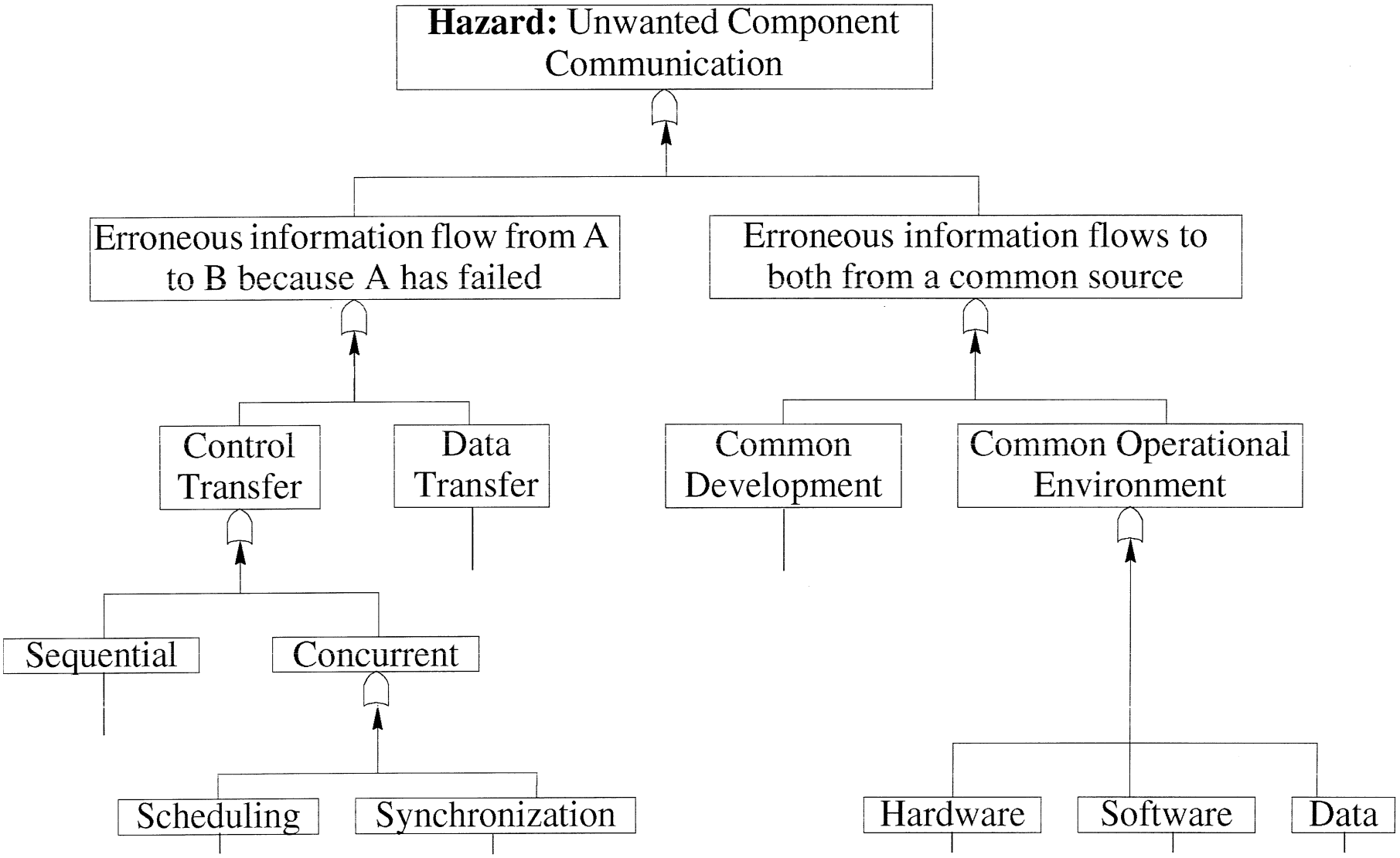
UVA
Department of Computer Science

COMPONENT MODEL OF SW DEPENDABILITY



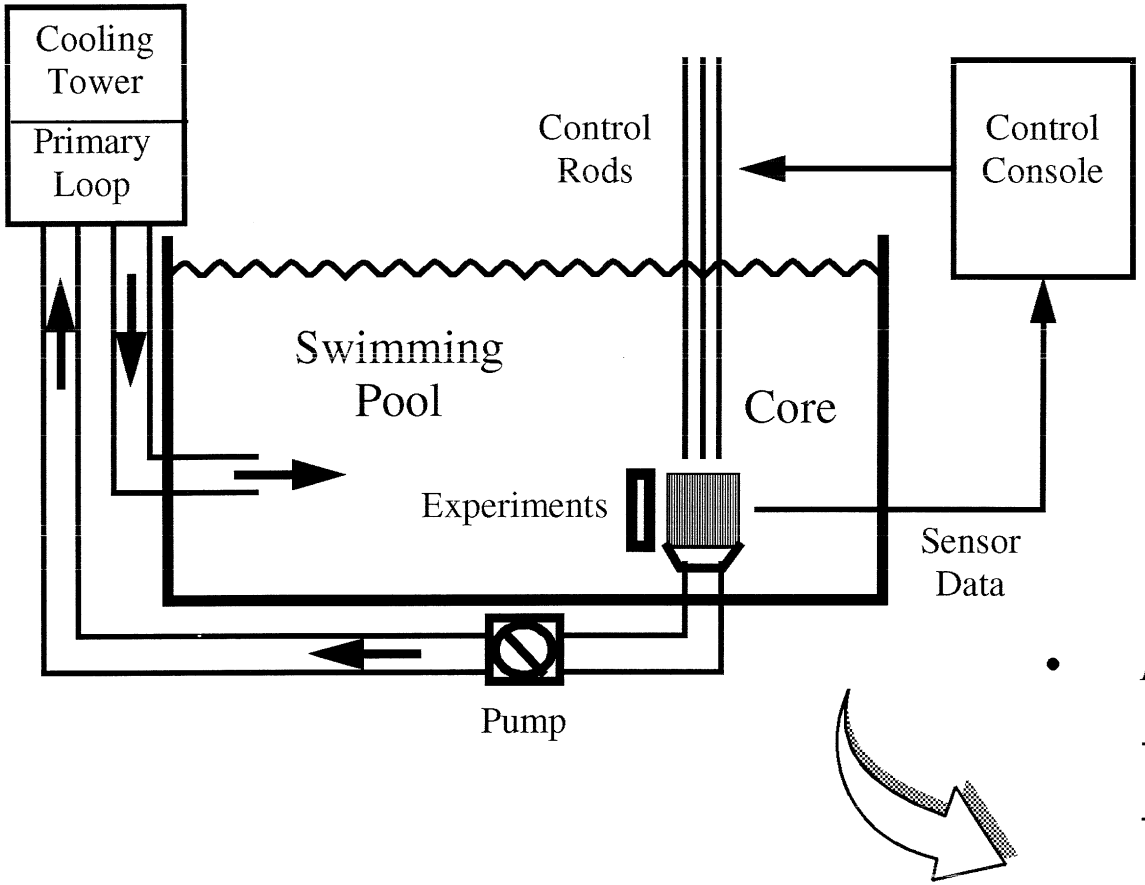
UVA
Department of Computer Science

COMPONENT INTERACTION MODEL



UVA
Department of Computer Science

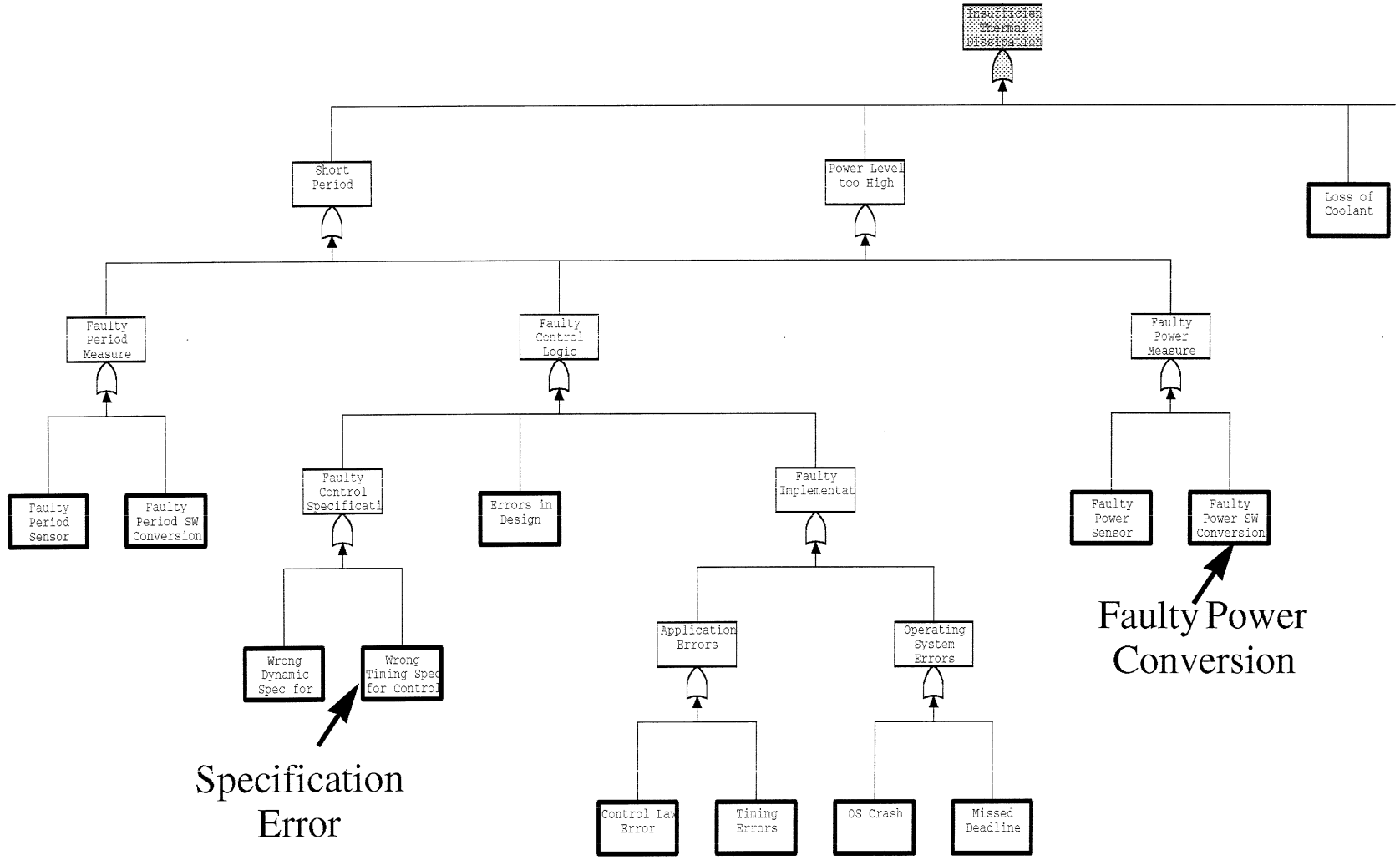
UNIVERSITY OF VIRGINIA REACTOR SYSTEM



- *Lots* Of Software:
  - Operating Systems
  - Network
  - Window System
  - User Interface

UVA
Department of Computer Science

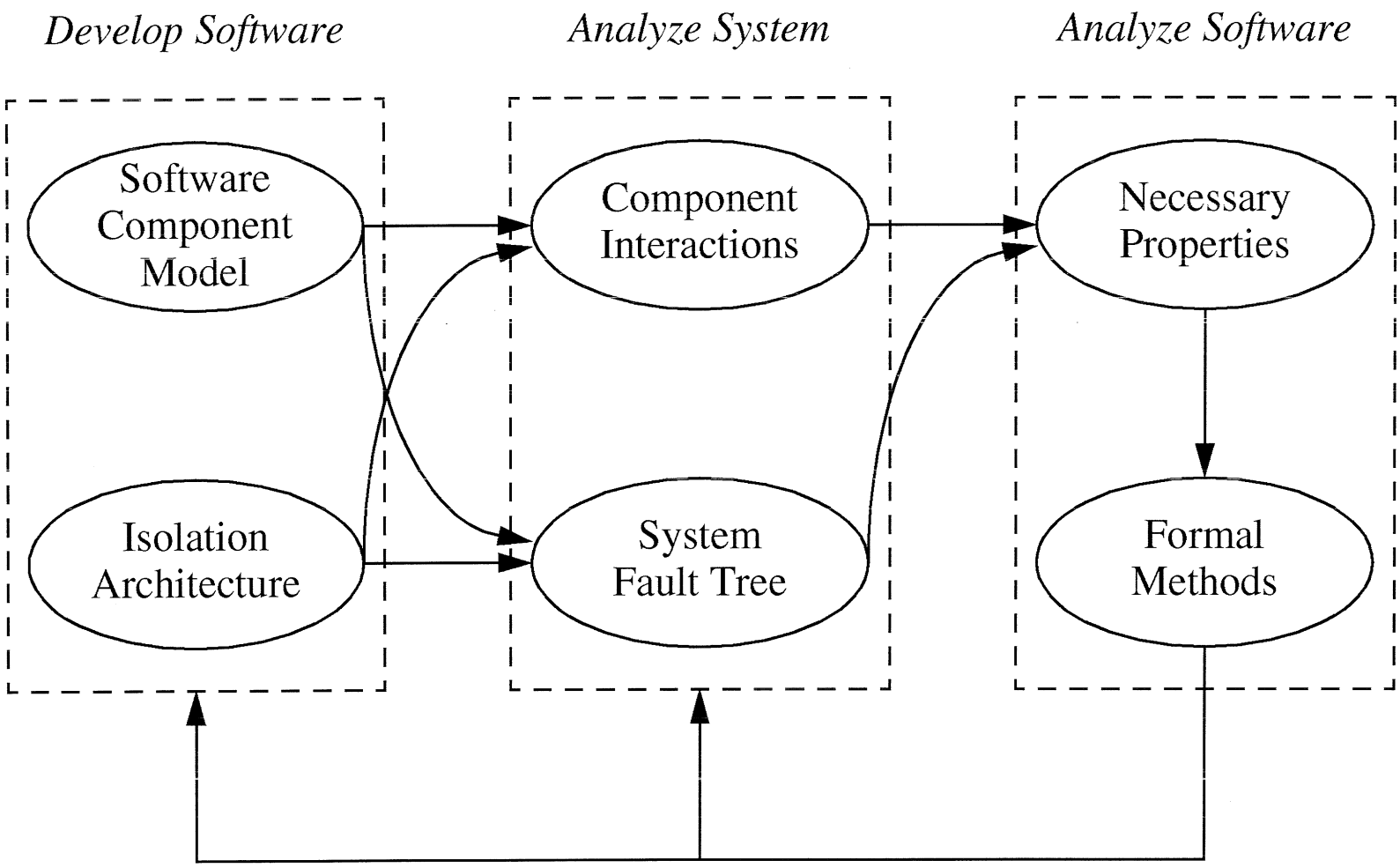
SOFTWARE FAULTS



UVA

Department of Computer Science

INTEGRATED ROLE OF FORMAL TECHNIQUES



UVA
Department of Computer Science

CONCLUSIONS

- Software Not Well-Integrated With System Dependability Assessment
- No Precise Role For Formal Techniques In Development
- Formal Techniques Don't Contribute Directly To System Dependability Assessment
- Component Model Of Software Dependability Developed
- Based On:
  - Component Software Design
  - Component Interaction Model
  - Integration Of Component Analysis In System Fault Tree
  - Enhancement Of Software Design Based On Fault-Tree Analysis